ANL/ALCF/ESP-17/4

Argonne NATIONAL LABORATORY

Flow, Mixing and Combustion of Transient Turbulent Gaseous Jets in Confined Cylindrical Geometries

Technical Report for the ALCF Theta Early Science Program

Argonne Leadership Computing Facility

ALCF Early Science Program (ESP) Technical Report

ESP Technical Reports describe the code development, porting, and optimization done in preparing an ESP project's application code(s) for the next generation ALCF computer system. This report is for a project in the Theta ESP, preparing for the ALCF Theta computer system.

ANL/ALCF/ESP-17/4

Flow, Mixing and Combustion of Transient Turbulent Gaseous Jets in Confined Cylindrical Geometries

Technical Report for the ALCF Theta Early Science Program

edited by
Timothy J. Williams and Ramesh Balakrishnan

Argonne Leadership Computing Facility

prepared by
Christos E. Frouzakis, George K. Giannakopoulos, Mahmoud Jafargholi, Stefan G. Kerkemeier, Misun Min, Ananias G. Tomboulides, Paul F. Fischer, and Scott Parker

September 2017

This work was supported in part by the Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Flow, Mixing and Combustion of Transient Turbulent Gaseous Jets in Confined Cylindrical Geometries

Christos E. Frouzakis[1], George K. Giannakopoulos[1], Mahmoud Jafargholi[1], Stefan G. Kerkemeier[1], Misun Min[2], Ananias G. Tomboulides[3,2], Paul F. Fischer[4,2], and Scott Parker[5]

[1] Aerothermochemistry and Combustion Systems Laboratory (LAV), Swiss Federal Institute of Technology (ETH), Zurich, Swizerland
[2] Mathematics and Computer Science Division, Argonne National Laboratory
[3] Laboratory of Applied Thermodynamics, Department of Mechanical Engineering, Aristotle University of Thessaloniki, Greece
[4] Depts. of Computer Science and Mechanical Science & Engineering University of Illinois at Urbana Champaign
[5] Leadership Computing Facility, Argonne National Laboratory, Argonne, IL

1 Introduction

Limitations in both experimental and modeling approaches inhibit the investigation of mixing and combustion in transient reacting jets. Even with heavy experimental effort, temporally resolved measurements to simultaneously assess the flow, mixing and combustion face significant difficulties in resolving the multitude of fast and spatially distributed variables (velocity, temperature and composition of a large number of chemical species) in large domains with spatial resolution sufficient to resolve the smallest scales.

Direct numerical simulation (DNS) can provide the complete description of the system state fully resolved in space and time. The accurate simulation of laboratory-scale setups using sufficiently detailed descriptions of the combustion kinetics is conditioned upon the availability of high performance computing (HPC) architectures and efficient numerical tools that can harness their computational power. Our reactive flow solver is built as a plugin to Nek5000 [1] and has been used in different HPC clusters to simulate flow, mixing and combustion phenomena in the gas-phase as well as on catalytic surfaces in laboratory-scale setups.

The project was selected as Tier 2 and received allocations mainly for porting, tuning, and debugging the solver on the KNL architecture. Code development also focused on the implementation of new algorithms to extend the modeling capabilities of the code for internal combustion engine applications.

2 Science Summary

The long-term aim of the project in the production runs is to simulate transient reacting turbulent jets, which are directly relevant for novel internal combustion engine concepts. Compared to steady

jets which have been extensively studied using experimental, theoretical and numerical approaches, transient jets have received less attention. This is mainly due to the fast, transient and interacting processes during the different injection phases, which can be captured experimentally only with demanding instrumentation. The fuel of interest is methane, the main component of natural gas, a low carbon fuel, which, thanks to its high H/C ratio produces about 25% less CO2 per thermal energy unit than oil products consisting of higher hydrocarbons.

3 Codes, Methods and Algorithms

The direct numerical simulation code is based on the open source spectral element solver Nek5000 developed at ANL [1] and is complemented by the plugin developed at the Aerothermochemistry and Combustion Systems Laboratory (LAV) at ETH Zurich for the energy and chemical species governing equations for combustion at the low Mach number limit [2]. The reactive flow solver can account for detailed gas-phase chemistry and transport properties as well as for detailed surface kinetics (catalytic combustion) in complex and variable geometries.

The governing partial differential equations are discretized in space using the spectral element method (SEM), a high-order weighted residual approach offering not only the rapid convergence and tensor-product efficiency of global spectral methods but also the geometric flexibility of finite elements [3]. The computational domain is decomposed into E conforming smaller subdomains (elements), which are quadrilaterals (in 2-D) or hexahedra (in 3-D) that conform to the domain boundaries. Within each element functions are expanded as N^{th}-order polynomials cast in tensor-product form, which allows differential operators on N^3 gridpoints per element to be evaluated with only $O(N^4)$ work and $O(N^3)$ storage; typically $N = 8 - 16$.

Nek5000 uses locally-structured basis coefficients ($N \times N \times N$ arrays), which allow direct addressing and tensor-product-based derivative evaluation that can be cast as efficient matrix-matrix products (MxM) involving N^2 operators applied to N^3 data values for each element. Correctly implemented, the data movement per gridpoint is the same as for finite-difference methods. The high polynomial representations yield minimal numerical dispersion and dissipation. At the same time, the increased computational intensity, $O(N)$ operations per memory reference, incurs no overhead in data movement and is thus ideally suited to modern architectures where the cost of data movement is dominant. For turbulent flow calculations at a given accuracy, Nek5000 has been shown to reduce the number of gridpoints and the computational cost (CPU hours) by an order of magnitude over standard low-order methods [4].

Temporal discretization is based on a high-order splitting that is 3rd-order accurate in time and reduces the coupled velocity-pressure Stokes problem to four independent elliptic solves per timestep: one for each velocity component and one for the pressure. The velocity problems are diagonally dominant and thus easily solved using Jacobi-preconditioned conjugate gradient iteration. The pressure substep requires a Poisson solve at each step, which is effected through multigrid-preconditioned GMRES iteration coupled with temporal projection to find an optimal initial guess. Particularly important components of Nek5000 are its scalable coarse-grid solvers that are central to parallel multigrid. The code features a fast direct solver for small-scale production runs (up to roughly 100,000 elements) that is optimal up to processor counts of $P \approx 10^4$–10^5, and fast algebraic multigrid for $P = 10^5$ and beyond. Counts of 15 GMRES iterations per timestep for billion-gridpoint problems are typical with the current pressure solver.

The reactive flow plugin is based on a high-order splitting scheme [5], where the continuity and momentum equations are integrated with the semi-implicit scheme of Nek5000, whereas the species and energy equations are integrated implicitly using the stiff integrator CVODE from the SUite of Nonlinear and DIfferential/ALgebraic equation Solvers (SUNDIALS). Recently, the Arbitrary Lagrangian/Eulerian (ALE) approach was extended to the low-Mach formulation to account for the mesh variation resulting from piston movement, allowing the study of compressed turbulence with varying thermodynamic pressure, which is the case of internal combustion engines (ICEs). This development has facilitated of large scale DNS in ICE geometries for gas-exchange calculations at ETH Zurich [6, 7, 8].

4 Code Development

4.1 New Methods/Algorithms

The aforementioned first ICE direct numerical simulations considered a fixed-valve geometry where only the piston was moving. The strong distortion of the grid resulting from moving valves results in higher computational costs and increased numerical error. Code development during this project included the non-trivial extension to more general geometries with moving valves through

- development of efficient mesh deformation algorithms and workflows (creation of new meshes offline allowing for layer addition, removal and remeshing and efficient and accurate spectral interpolation of the data from the old to the new mesh) suitable for complex geometries with moving piston and valves,
- extension of the characteristics method (OIFS) for efficient time integration with time-varying geometries to bypass standard Courant-Friedrichs-Lewy (CFL) constraints,
- efficient coupling of the above with the low-Mach-number reactive flow solver,
- on the fly high order scalable grid-to-grid interpolation to enable the use of multiple meshes during the simulation.

4.2 Performance and optimization on Theta

The code performance was studied in both cold and reactive setups. In the former case, the incompressible turbulent flow in a channel was considered first is a small case with $E = 512$ spectral elements and polynomial order $N = 7$ that could fit on a single KNL node. Our scaling studies on Theta showed a 7.2× per node speedup in comparison to BG/Q (Fig. 1(a)). Nek5000 scales well on Theta and can use all cores on a node using pure MPI.

Very good strong scaling is also exhibited across nodes with 73% parallel efficiency for as few as 13,500 points per MPI rank (corresponding to 36,000 degrees of freedom, Fig. 1(b)), and is the lower limit of typical production runs. Due to its low memory footprint, Nek5000 can often fit into the high-bandwidth on-package memory (MCDRAM), and therefore no code modifications related to memory management was needed for the new architecture.

The code performance on Theta was tested with both the cache-quad and the flat-quad memory modes with numactl-m1, but no noticeable difference was observed. Similarly, the use of hardware

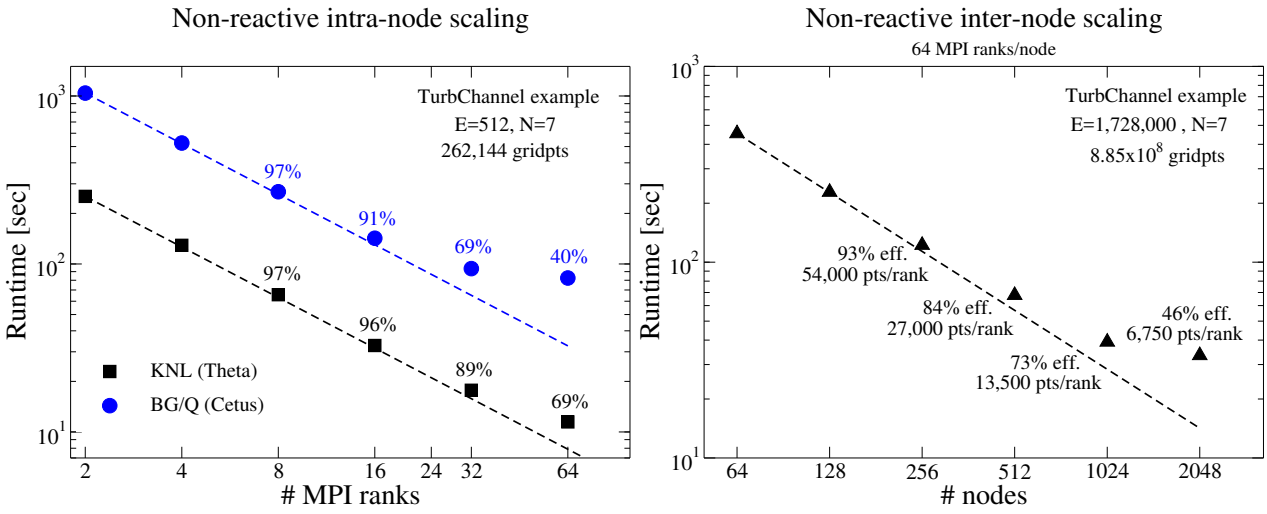


Figure 1: Strong scaling of an incompressible turbulent channel flow simulation on Theta and Mira: (a) single-node performance, (b) performance across nodes.

threads did not result in performance improvement since the memory bandwidth is saturated by using one rank per physical core.

In the reactive simulations, the species production rates and the thermodynamic and transport properties are by default computed using the Chemkin package [9, 10], which provides general-purpose subroutines to easily account for the typically large number of species and chemical reactions needed to accurately describe the complex combustion kinetics. By instrumenting the code with timers, the hotspots of the reactive solver in a benchmark problem of a laminar planar premixed methane-air flame propagation in a 3-D rectangular domain were identified as the subroutines computing the reaction rates (84%), the MxM routines (3.8%), the species transport properties (2.6%), and the pressure solver (2.3%). A skeletal mechanism consisting of 20 chemical species in 97 reactions that was derived from the detailed GRI-3.0 scheme using the entropy analysis method [11] was used.

Aiming at simulations with more complex fuels, where the cost for chemical source terms and transport properties will increase with the number of species, a matlab-based code was developed that uses the standard Chemkin input files to generate optimized fuel-specific subroutines for the thermochemistry and transport properties kernels. Figute 2 shows that a speedup of 5.4x and 3.2x was achieved in the evaluation of the reaction rates and the transport properties, respectively. Overall, a 3.6x speedup of the total time-to-solution over the Chemkin subroutines was reached.

After optimization of the chemical source term and transport properties, the hotspot contributions to the overall cost changed to: reaction rates 56% , transport 2.9%, MxM 13%, and pressure solve 4.8%. Further reduction in the time spent on the thermochemistry subsystem can be achieved by fine tuning the optimized routines manually to exploit all vectorization capabilities. Initial tests showed an additional 3x performance improvement over the optimized mechanism for hydrogen combustion (9 species, 21 reactions). Since the local work in the reactive flow solver increases with the molecular complexity of the fuel, larger gains are expected for hydrocarbons. A threaded version of the vectorized thermo-chemistry routines is currently under development. Current results indicate a 20% performance improvement using two hardware threads/core. The substantial overhead created by using four hardware threads/core resulted in performance that is lower than using a single thread

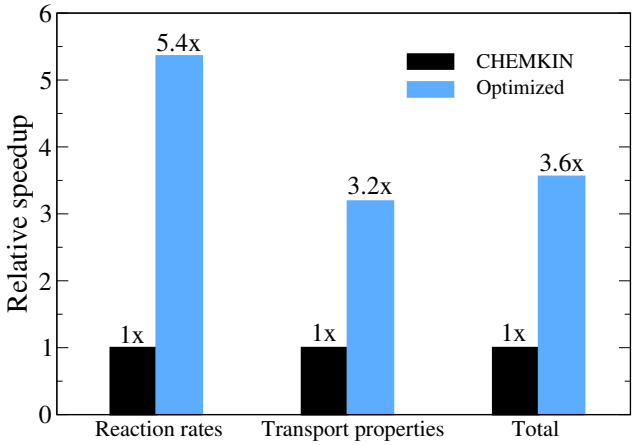


Figure 2: Chemistry optimization for a skeletal reaction mechanism for methane-air combustion containing 20 chemical species.

per core.

The intra-node performance of the code on the benchmark problem was analyzed on different architectures using the optimized routines. Very good intra-node performance was observed for the reactive flow simulations. The performance on a single KNL node is shown in Fig. 3(a) (black line with filled squares). The significantly higher parallel efficiency compared to the turbulent channel flow for approximately the same number of grid point per rank is due to the increased local work for the calculation of reaction rates, diffusion coefficients and thermodynamic properties. This allows for an even smaller number of elements/MPI rank for reactive runs, so that the problem can fit into the MCDRAM even with the larger mechanisms that are needed for more complex fuels.

Compared to other architectures, KNL was found to be faster by 12.5x than BG/Q and 1.3x slower than Broadwell (E5-2695 v4 2x18 cores on the Piz Daint system of the Swiss Supercomputing Center CSCS). It should however be noted that significant system noise was observed on Theta, which prevents accurate benchmarking. No notable speedup ($< 10\%$) was observed between cache-quad and flat-quad with numactl –membind=1 modes. In flat-quad mode, a speedup of 3x was obtained when using MCDRAM (–membind=1) instead of DDR (–membind=0).

The reactive code performance on up to 3,072 nodes of Theta is reported in Fig. 3(b). An efficiency of 75% with only 5,120 pts/rank (corresponding to 3,400 unique grid points) was observed using more than 80% of the machine. This shows that typical production runs will make very efficient use ($> 90\%$) of the system. More importantly, it makes the code suitable for the ALCF capability job scheduling strategy, where large jobs with short wall-times are favored over small and long ones.

In order to reduce the time taken by the matrix-matrix multiplications (MxM), the code was also coupled to the *libxsmm* library, which incorporates a just-in-time compiler to optimize for tensor contractions cast as matrix-matrix products. For the chosen polynomial order the computational cost benefits for the reactive solver over the tuned MxM kernel of Nek500 were minimal.

The use of threads was investigated using Nekbone mini-app, which captures the basic structure and user interface of Nek5000 so that the computationally intensive linear solvers accounting for a large percentage of the cost of Nek5000, as well as the communication costs required for nearest-neighbor data exchanges and vector reductions can be effectively analyzed. No significant benefit was found

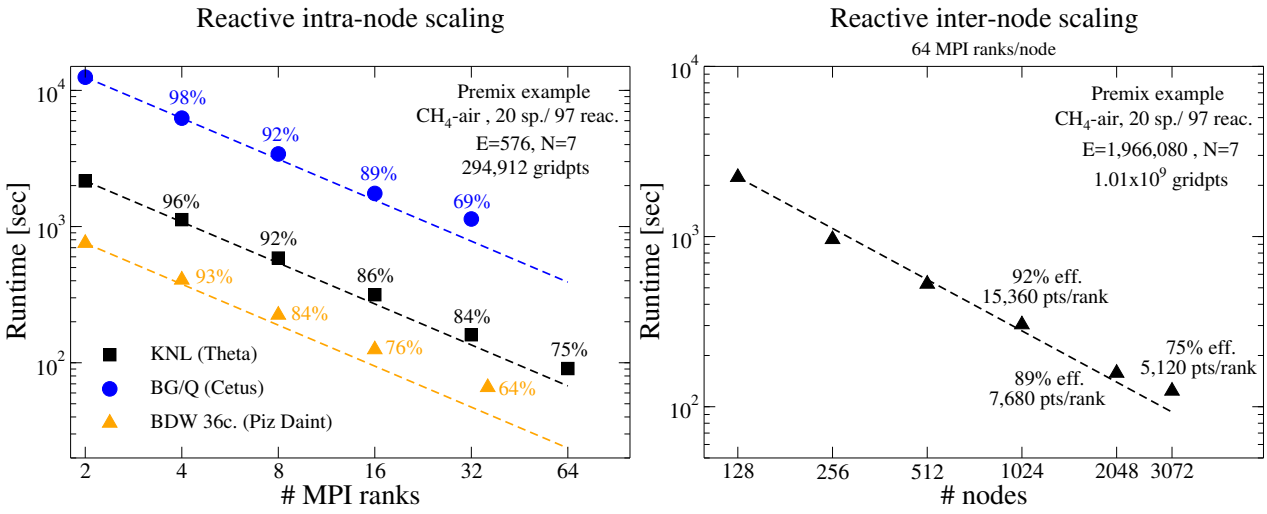


Figure 3: Strong scaling of a methane/air premixed flame propagation simulation Theta and Mira: (a) single-node performance, (b) performance across nodes.

on KNL for the considered number of total MPI ranks, since pure MPI (32,768 ranks $\times$ 1 thread) and MPI+OpenMP (4,096 $\times$ 8) were found to provide essentially identical performance.

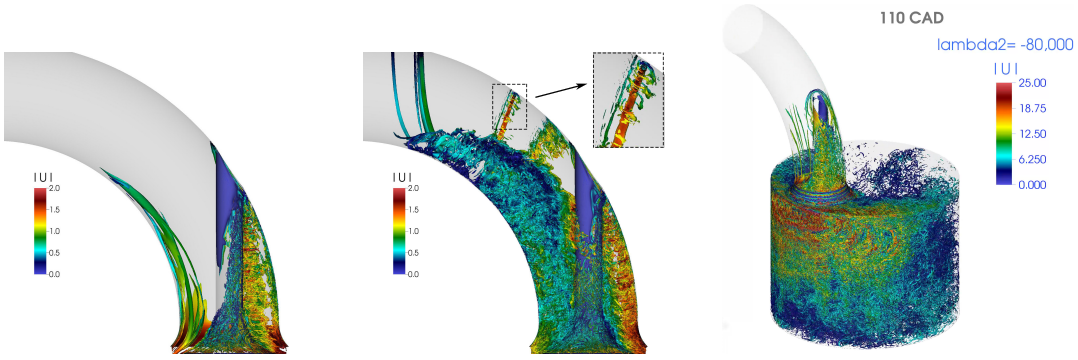
4.3 Applications

Part of the allocated time was used to perform direct numerical simulations with the new capabilities introduced in the code. The main focus was on flow phenomena in the TCC-III engine [12] and on flow-flame and flame-wall interactions during the ignition of the fuel in the main chamber by the hot reactive gases generated in a prechamber.

4.3.1 Internal combustion engines

The first step towards the simulation of the whole TCC-III engine was to study the effect of the inflow velocity profile on the flow in the intake duct at realistic Reynolds numbers. The results were presented in the 11th International ERCOFTAC Symposium on Engineering Turbulence Modeling and Measurements (ETMM), 21-23 September, Palermo, Italy and will appear in [13]. Typical flow structures resulting from the flow inside the curved pipe and around the valve stem are shown in Figs. 4(a)-(b).

Subsequently, the intake pipe was coupled with the full engine geometry and the intake stroke was simulated with the valve fixed at its halfway-open position. The rich vortical structure inside the cylinder can be seen in Fig. 4(c). Current simulations make use of the code developments in the frame of this project to perform complete multi-cycle simulations accounting for the movement of the inflow and exhaust valves.



(a) (b) (c)

Figure 4: Instantaneous vortical structures identified by the $\lambda_2 = -15$ isosurfaces colored with the flow velocity magnitude for simulations with (a) power-law and (b) parabolic inflow velocity profile at $Re = 33,000$; (c) vortical structures at 110 crank angle degrees after top dead center.

4.3.2 Pre-chamber ignition

In the frame of a study of pre-chamber ignition at LAV, we are simulating a hot, transient reacting jet generated by igniting a stoichiometric methane/air mixture in a small combustion chamber attached to a main chamber that contains a difficult to ignite lean mixture.

5 Portability

The code conforms to the FORTRAN and C ANSI standards and employs the MPI standard for parallelism. Nek5000 is essentially a stand-alone code, and the stiff ODE integrator CVODE employed by the reactive flow solver compiles without problems on different systems. Because of its scalability and minimal dependencies, Nek5000 has long been a first-deployed simulation code on leading-edge supercomputers. It runs with or without MPI on platforms ranging from notebooks to Linux clusters to supercomputers (IBM BG/P, BG/Q and Cray systems) using different compilers (GNU, PGI, IBM, Cray, Intel). Over 800 integral tests, with four different compilers, are ran after each code update to ensure continued portability. The incompressible as well as the reactive flow codes compile and run efficiently with minor modifications on Theta.

References

[1] P.F. Fischer, J.W. Lottes, and Kerkemeier S.G. nek5000 Web page, 2008. http://nek5000.mcs.anl.gov.

[2] S.G. Kerkemeier. *Direct numerical simulation of combustion on petascale platforms: application to turbulent non-premixed hydrogen autoignition.* PhD thesis, Swiss Federal Institute of Technology Zurich, Nr. 19162, 2010.

[3] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-order methods for incompressible fluid flow.* Cambridge University Press, 2002.

[4] M. Sprague, M. Churchfield, A. Purkayastha, P. Moriarty, and S. Lee. A comparison of Nek5000 and OpenFOAM for DNS of turbulent channel flow. `http://nek5000.mcs.anl.gov/index.php/Usermeeting2010`, 2010.

[5] A.G. Tomboulides, J.C.Y Lee, and S. Orszag. Numerical simulation of low mach number reactive flows. *J. Sci. Comp.*, 12(2):139–167, 1997.

[6] M. Schmitt, C.E. Frouzakis, Y.M. Wright, A. Tomboulides, and K. Boulouchos. Direct numerical simulation of multiple cycles in a valve/piston assembly. *Phys. Fluids*, 26(3):035105, 2014.

[7] M. Schmitt, C.E. Frouzakis, Y.M. Wright, A. Tomboulides, and K. Boulouchos. Investigation of cycle-to-cycle variations in an engine-like geometry. *Phys. Fluids*, 26(12):125104, 2014.

[8] M. Schmitt, C.E. Frouzakis, Y.M. Wright, A. Tomboulides, and K. Boulouchos. Direct numerical simulation of the compression stroke under engine relevant conditions: Local wall heat flux distribution. *Int. J. Heat Mass Tansfer*, 92:718–731, 2015.

[9] R.J. Kee, F.M. Rupley, and J.A. Miller. Chemkin II: a Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. Technical Report SAND89-8009B, Sandia National Laboratories, 1989.

[10] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, and J.A. Miller. A Fortran computer code package for the evaluation of gas-phase multicomponent transport properties. Technical Report SAND86-8246, Sandia National Laboratories, 1986.

[11] M. Kooshkbaghi, C.E. Frouzakis, Boulouchos K, and I. Karlin. Entropy production analysis for mechanism reduction. *Combust. Flame*, 161(6):1507–1515, 2014.

[12] P. Schiffmann, S. Gupta, D. Reuss, V. Sick, X. Yang, and T.-W. Kuo. TCC-III engine benchmark for large-Eddy simulation of IC engine flows. *Oil & Gas Sci. Techn. - Rev. IFP Energies Nouvelles*, 71(3):1–27, 2016.

[13] G.K. Giannakopoulos, C.E. Frouzakis, K. Boulouchos, P.F. Fischer, and A.G. Tomboulides. Direct numerical simulation of the flow in the intake pipe of an internal combustion engine. *Int. J. Heat Fluid Flow*, in press, 2017.

Argonne Leadership Computing Facility
Argonne National Laboratory
9700 South Cass Avenue, Bldg. #240
Argonne, IL 60439

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC